

Arhitectura Sistemelor de Calcul (ASC)

Examinarea finală

Varianta 2

(2025 - 2026)

Anul I, Semestrul I
ianuarie/februarie 2026
Cristian Rusu

Completați aici totul cu majuscule.

Nume: _____

Prenume: _____

Grupa: _____

Toate răspunsurile sunt în albastru.

Înainte de a începe, citiți cu atenție indicațiile următoare:

- Testul și rezolvarea sa vor fi disponibile online în zilele următoare.
- *Nu aveți voie cu laptop-uri sau alte dispozitive de comunicație.*
- *Nici calculatoarele de buzunar nu sunt permise.*
- *Vă rugăm să vă opriți telefoanele mobile.*
- Pentru întrebările cu răspunsuri multiple/simple folosiți tabelele puse la dispoziție.
- Acest test are 6 enunțuri totalizând 100 de puncte.
- Aveți la dispoziție 120 de minute pentru a completa examinarea.
- Mult succes!

Întrebarea 1. (22 puncte)

Completați tabelul de mai jos cu valorile numerice corecte. Toate numerele sunt naturale pe 16 biți. Opriți calculele la 16 biți. (Fiecare răspuns corect valorează 1 punct)

binar	octal	zecimal	hexazecimal
b00000111101111010111	o075727	31703	0x7BD7
b00000001101111111101	o015775	7165	0x1BFD
b00000010100111101100	o024754	10732	0x29EC
b00000111011011101111	o073357	30447	0x76EF

a binar	a hexa	b binar	b hexa	a+b zecimal	a+b binar	a+b hexa
b0110111100110101	0x6F35	b0101011011111100	0x56FC	50737	b1100011000110001	0xC631

a binar	a hexa	b binar	b hexa	axb zecimal	axb binar	axb hexa
b0111010110111011	0x75BB	b0111101111001111	0x7BCF	61834	b1111000110001010	0xF18A

Răspunsurile care țin cont de overflow sunt de asemenea corecte.

Întrebarea 2. (13 puncte)

Completați tabelul de mai jos cu valorile numerice corecte. Toate numerele sunt întregi pe 12 biți. (Fiecare răspuns corect valorează 1 punct)

binar	octal	zecimal	hexazecimal
b110000101010	o6052	-982	0xC2A
b101110001000	o5610	-1144	0xB88

a zecimal	a binar	a hexa	b zecimal	b binar	b hexa
-2046	b100000000010	0x802	-16	b111111110000	0xFF0

produs axb zecimal	produs axb binar	produs axb hexa
32736	b011111111100000	0x7FE0

Răspunsurile care țin cont de overflow sunt de asemenea corecte. S-au punctat și cazurile în care octal și hexazecimal s-au considerat doar 10 biți (1 în loc de F sau 1 în loc de 7).

Întrebarea 3. (17 puncte)

Răspundeți la următoarele întrebări scurte. Completați în tabelul de jos.

- 3.1. (2 puncte) Cine a scris primul compilator COBOL?
- 3.2. (2 puncte) Care este valoarea în zecimal a unui număr care în octal este un șir de n cifre 7?
- 3.3. (2 puncte) Simplificați maxim expresia logică $B+(A!C+AB)+(A!B+AC)+A!B+!A!B$.
- 3.4. (2 puncte) În formatul IEEE Float 32 biți ce număr, în hexazecimal, este înaintea numărului 0x40000000?
- 3.5. (2 puncte) Scrieți în formatul IEEE Float 32 biți, hexazecimal, numărul zecimal -183.125.
- 3.6. (2 puncte) Ce hazard produce următoarea secvență de cod? add t1, t2, t3; add t1, t4, t5.
- 3.7. (2 puncte) Vi se dă un număr natural a . Vrem să extragem *lungime* începând cu poziția *start* (de la dreapta către stânga) – adică vrem șirul $0 \dots 0a_{(\text{start} + \text{lungime})} \dots a_{\text{start}}$. Cum faceți? Simplificați maxim expresia logică.
- 3.8. (3 puncte) Care este numărul maxim de biți folosit pentru codarea eficientă a unui caracter din mesajul entropieshannon?

3.1.	Grace Hopper
3.2.	$8^n - 1$
3.3.	1
3.4.	0X3FFFFFFF
3.5.	0xC3372000
3.6.	WAW
3.7.	$(a \gg \text{start}) \& ((1 \ll \text{lungime}) - 1)$
3.8.	4 (cu Huffman)

Întrebarea 4. (16 puncte)

Ați ajuns să lucrați la celebra companie Mvidia. Trebuie să decideți baza de calcul pentru noile procesoare, dar din păcate ați adormit la curs de ASC când s-au discutat bazele. Într-un moment neinspirat, alegeți baza B în felul următor: dacă vi se dau n triți $t_{n-1}t_{n-2}\dots t_2t_1t_0$ valoarea în baza 10 este $x = \sum_{i=0}^{n-1} t_i 2^i$, $t_i \in \{-1, 0, 1\}$. Răspundeți la următoarele întrebări:

- 4.1. (2 puncte) Pentru $n = 4$ care sunt valorile, în baza 10, pe care acest sistem le poate reprezenta?
- 4.2. (2 puncte) Care este cel mai mare număr care poate fi scris în acest fel pe n biți (triți), n este par? Dar cel mai mic? Pentru ambele cazuri dați valoarea în baza 10. Câte combinații de valori se pot scrie pe n biți și câte dintre acestea sunt unice?
- 4.3. (6 puncte) Vi se dă un număr scris în această bază, scrieți numărul în complement față de doi. Răspunsul rămâne pe n biți, orice overflow este ignorat.
- 4.4. (6 puncte) Se dau două numere în această bază, explicați cum le adunați bit cu bit.

4.1. Valorile sunt de la -15 la 15.

4.2. Cel mai mic număr este $-2^n + 1$ iar cel mai mare este $2^n - 1$

Sunt în total 3^n combinații posibile dar doar $2 \times 2^n - 1 = 2^{n+1} - 1$ valori distincte, 2^n pozitive și 2^n negative iar 0 este numărat de două ori.

4.3. Definim două măști de biți: masca p astfel încât $p_i = 1$ acolo unde $t_i = 1$ și masca m astfel încât $m_i = 1$ acolo unde $t_i = -1$. Răspunsul este $p - m$ dar avem $(p - m) \& (2^n - 1)$ ca să păstrăm ultimii n biți doar.

4.4. Când adunăm 3 biți a_i , b_i și c_{in} vom avea rezultatele s_i și c_{out} . Definim $t = a_i + b_i + c_{in}$ și vom afla s_i și c_{out} astfel încât $t = s_i + 2c_{out}$. Aici $t \in \{-3, -2, -1, 0, 1, 2, 3\}$. Soluțiile sunt: i) dacă $t \in \{-1, 0, 1\}$ atunci $s_i = t$ și $c_{out} = 0$; ii) dacă $t = 3$ atunci $s_i = 1$ și $c_{out} = 1$; iii) dacă $t = 2$ atunci $s_i = 0$ și $c_{out} = 1$; iv) dacă $t = -2$ atunci $s_i = 0$ și $c_{out} = -1$; v) dacă $t = -3$ atunci $s_i = -1$ și $c_{out} = -1$.

De unde vine $t = s_i + 2c_{out}$? Vine de la ecuația $(a_i + b_i + c_{in})2^i = s_i 2^i + c_{out} 2^{i+1}$, este adunarea care se întâmplă pe poziția i (plus carry-ul transmis la poziția $i + 1$). Ecuația asta ne spune cât carry este mereu ca rezultatul total al adunării să nu se modifice. Ca rezultatul să fie corect, această ecuație trebuie să fie satisfăcută pe tot parcursul adunării bit cu bit.

Astfel de sisteme de numărare se numesc *balanced binary*.

Întrebarea 5. (15 puncte)

Ca să găsim soluțiile unei ecuații de grad doi $ax^2 + bx + c = 0$ vom folosi bine-cunoscutele formule: $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. În 2019 cercetătorul Po-Shen Loh, a găsit o nouă modalitate ingenioasă de a organiza calculele pentru a calcula $x_{1,2}$ urmând algoritmul:

1. Ecuația trebuie să fie de forma $x^2 + bx + c = 0$, (adică $a = 1$).
2. Calculează $m = -b/2$.
3. Calculează $u = \sqrt{m^2 - c}$.
4. Soluțiile sunt $x_{1,2} = m \pm u$.

Răspundeți la următoarele cerințe:

- 5.1. (2 puncte) Rezolvați ecuația $x^2 - x + 6 = 0$ (calculați explicit m , u și apoi $x_{1,2}$).
- 5.2. (2 puncte) Rezolvați ecuația $2x^2 - 12x + 10 = 0$ (calculați explicit m , u și apoi $x_{1,2}$).
- 5.3. (4 puncte) Completați tabelul de mai jos cu numărul de operații aritmetice. Presupunem că toate înmulțirile/împărțirile cu 2 și 4 se fac cu operații de deplasare.

	+	$\times$	/	$\sqrt{}$	$\gg, \ll$
original	4	2	2	1	2
Loh cu $a = 1$ de la început	4	1	0	1	1
Loh cu $a \neq 1$ de la început	4	1	2	1	1

- 5.4. (2 puncte) Care variantă o preferați între cea originală și Loh cu $a \neq 1$?
- 5.5. (5 puncte) Dacă sistemul de calcul este capabil de *out of order execution* atunci verificați care variantă de calcul este preferată (număr mai mic de stagii/pași de calcul) între varianta originală și Loh cu $a = 1$ de la început.

5.1. $m = 1/2$, $u = i\sqrt{23}/2$ și $x_{1,2} = 1/2 \pm i\sqrt{23}/2$.

5.2. Ca să aplicăm noile formule trebuie să avem $a = 1$ deci împărțim ecuația cu 2 ca să obținem $x^2 - 6x + 5 = 0$ și apoi rezolvăm: $m = 3$, $u = 2$ și $x_{1,2} = \{5, 1\}$.

5.4. Loh pentru că are mai puține înmulțiri și deplasări.

5.5. Conform Cursului 0x08 rezolvarea originală are 6 stagii/pași de calcul. Pentru Loh avem: 1) $m = -b$, 2) $m = m/2$, 3) $t = m^2$, 4) $t = t - c$, 5) $t = \sqrt{t}$ și 6) $t_{1,2} = m \pm t$. Deci tot 6 stagii/pași.

Se punctează și dacă ați numărat puțin diferit dar justificat (sau dacă ați considerat și cazul cu soluții complexe, unde sunt mai puține adunări pentru că $1 + i$ nu e adunare). Dacă $a \neq 1$ atunci sunt două împărțiri la Loh. Punctele 5.1 și 5.2 erau să vedeți aceste lucruri explicit.

Notă: Lucrarea originală a lui Loh o găsiți la <https://arxiv.org/abs/1910.06709> În realitate, presupunerea că operațiile cu constantele 2 și 4 devin deplasări pe biți este excesiv de optimistă.

Întrebarea 6. (20 puncte)

Trebuie să simulați legi fizice pentru N particule. Pentru fiecare particulă aveți: x, y, z, vx, vy, vz (toate float32). Pentru fiecare particulă de la 1 la N , în fiecare pas: $x += vx*dt, y += vy*dt, z += vz*dt$, unde dt este un pas de timp. Considerați o linie de cache de 64 bytes.

- 6.1. (4 puncte) Enumerați două probleme de performanță dacă folosiți un vector de structuri x, y, z, vx, vy, vz și iterați secvențial. Adică informațiile pentru fiecare particulă sunt contigue în memorie iar particulele sunt în ordine.
- 6.2. (8 puncte) Propuneți soluții pentru a îmbunătăți accesul la cache și vectorizarea.
- 6.3. (2 puncte) Câte particule încap într-o linie de cache dacă folosiți un vector de structuri? Câți bytes rămân nefolosiți?
- 6.4. (6 puncte) Explicați (scurt, dar tehnic) dacă puteți folosi SIMD în acest context.

6.1. Încărcări necontigue pentru x, y, z și vx, vy, vz . Nu se poate aplica SIMD din cauza asta. Cache miss mare.

6.2. În loc să grupăm informația pentru fiecare particulă, vom grupa toată informația în vectori x, y, z și vx, vy, vz . Păstrarea dt în registru, pentru că este utilizat în continuu.

6.3. Pentru fiecare particulă avem 6 floats = 24 bytes/particulă. În cache de 64 bytes: $\lceil 64/24 \rceil = 3$ particule (48 bytes). Spațiu nefolosit: 16 bytes.

6.4. Vectori de poziții și de viteze vor funcționa mai bine. În cache-uri de 64 bytes vor încăpea 8 poziții/viteze. SIMD pe 4 valori, cel puțin.